

АНАЛИЗ НА СОФТУЕРНИ АРХИТЕКТУРИ

ATAM & CBAM

Въведение и цели

- Досега беше разгледано проектирането и документирането на софтуерни архитектури
- Въпросът дали така проектираната и документирана архитектура ще доведе до система, която удовлетворява изискванията е въпрос на *Анализа на софтуерни архитектури*
- Ще разгледаме 2 примерни процеса за анализ на софтуерни архитектури
 - ATAM (Architecture Tradeoff Analysis Method)
 - CBAM (Cost Benefit Analysis Method)

Въведение в ATAM

- ATAM (Architecture Tradeoff Analysis Method) – метод за анализ, базиран на компромисите
- Разкрива до каква степен архитектурата удовлетворява индивидуалните качествени изисквания
- И по-важното – как архитектурните решения си взаимодействат и съответно какви компромиси се правят за това
- Анализът е сложен процес – обикновено големите системи имат сложна архитектура, многобройни бизнес цели и голям брой заинтересовани лица, а времето за анализ е твърде ограничено

Участници в АТАМ

- В процеса на анализ съгласно АТАМ участват:
 - **Оценяващ екип** – от 3 до 5 души, външни за проекта, чиято архитектура се оценява. Всеки от оценяващия екип изпълнява една или няколко роли
 - **Ръководен екип** – група, от която зависят решенията по проекта. Обикновено включва ръководителя на проекта, представител на клиента, архитекта и лицето, наредило анализа
 - **Останалите заинтересовани** – разработчици, тестери, поддръжка, потребители, разработчици на трети системи, интегратори и т.н.

Роли в оценяващия екип

- **Ръководител на екипа** – формира екипа, ръководи цялостния процес, установява необходимите контакти, координира действията с клиента и т.н.
- **Ръководител на анализа** – физически ръководи анализа, подпомага извличането на сценарии, администрира процеса на избор и приоритизация на сценариите и т.н.
- **Автор на сценарии** – формулира сценариите, използвайки съгласувана терминология! Изключително важна роля, може да спре дискусията докато не се постигне съгласие от всички по окончателната формулировка
- **Стенограф** – записва дискусиите колкото е възможно по-детайлно, вкл. документира окончателните формулировки

Роли в оценяващия екип

- **Хронометрист** – подпомага ръководителя на анализа с данни за това, колко отнема дискусията на всеки сценарии и може да прекратява дискусии при прекалено задълбаване
- **Наблюдател** – тихо наблюдава как върви процеса и предлага на ръководителя възможни подобрения, както за настоящия, така и за бъдещи анализи; в задълженията му влиза и да подготви отчет за проведения анализ и евентуално да предложи подобрения на процеса за анализ на архитектурата
- **Специалист по процеса** – знае ATAM наизуст и подпомага ръководителя да определя следващите стъпки
- **Разпитващ** – задава необходимите въпроси на всички участници

Резултати от ATAM

- Сбито представяне на архитектурата
 - обикновено архитектурата се отъждествява с обемисти списъци от обекти, интерфейси и сигнатури. ATAM води до създаването на сбито представяне, което да бъде демонстрирано в рамките на 1 час
- Изясняване на бизнес целите – често се случва разработчици да не са и чували за бизнес целите преди да се направи анализа
- Непосредствено се получават сценарии за по-важните изисквания към качеството (следствие от бизнес целите)
- Разглежда се съответствието между архитектурни решения и качествени изисквания

Резултати от АТАМ

- Идентифицират се решенията, към които качествените изисквания са чувствителни и които представляват компромиси
- Идентифицират се множествата от рискове и безопасни решения (nonrisks).
 - Риск – решение, което може да доведе до нежелани последствия
- Идентифицират се рискови тематики – проблемни области от архитектурата, където рисковете са систематични и чието присъствие неминуемо носи заплаха за проекта
- И други вторични резултати, вкл. и нематериални такива (напр. допълнителна документация, подобряване на комуникацията в екипа, по-добро разбиране на проекта от всички участници)

Четири фази на ATAM

- Фаза 0 е подготвителна – ръководствата на екипа за оценка и на ръководния екип се събират и уточняват всички подробности по анализа, напр. каква експертиза е необходима, логистиката, заинтересовани лица, план за изпълнение, налична документация и изисквания към представянията на ръководителя на проекта и на архитекта

Четири фази на ATAM

- Фази 1 и 2 са същинските фази на оценка – екипът по оценка вече е изучил наличната документация, има идея за системата, общата насоченост на архитектурата и т.н. Организиран се срещи с ръководния екип (на първа фаза) и с останалите заинтересовани лица (на втора фаза) и се осъществява същинския анализ

Четири фази на ATAM

- Фаза 3 е заключителна – подготвя се и се доставя окончателния доклад за състоянието на архитектурата. Друга важна дейност е самооценка на извършената работа и изводи за бъдещи проекти. Наблюдателя на процеса съставя съответния доклад за работата на екипа. След време (5-6 месеца) ръководителя на екипа се допитва до клиента с цел установяване на положителен (или отрицателен) резултат.

Участници и роли според фазите

Фаза	Дейности	Участници	Продължение
0	Подготовка	Ръководството на екипа по оценка и ключови участници от страна на клиента	Неформално, в продължение на няколко седмици
1 Стъпки 1—6	Оценка	Екипа по оценка и ръководния екип на проекта	1-2 дни, последвани от пауза от 2-3 седмици
2 Стъпки 7 – 9	Оценка 2	Екипа по оценка, ръководния екип и останалите заинтересовани лица	2-3 дни
3	Заключение	Екипа по оценка и клиента	1 седмица

Стъпка #1

- Стъпка #1 – представяне на метода. Ръководителя на екипа по оценка представя метода на всички участници във фазата, задават се въпроси, дават се отговори и се установява контекста и набора от дейности по време на процеса.

Стъпка #2

- Стъпка #2 – дискусия относно бизнес целите на системата. Ръководителя на проекта или представител на клиента представя системата от гледна точка на бизнеса:
 - Най-важните функции на системата;
 - Всякакви приложими технически, управленски, икономически и политически ограничения;
 - Бизнес целите и бизнес обкръжението и как те се отнасят към проекта;
 - Главните заинтересовани лица;
 - Архитектурните драйвери, т.е. най-важните качествени характеристики които оформят архитектурата

Стъпка #3

- Стъпка #3 – представяне на архитектурата. Архитекта, в рамките на 1 час (20-на слайда) представя архитектурата:
 - Основни изисквания, измерваеми величини свързани с тях, стандартни подходи за постигането им (2-3 слайда);
 - Важна архитектурна информация (4-8 слайда):
 - Контекстна диаграма – системата и нейното обкръжение (хора и системи, с които си взаимодейства);
 - Декомпозиция на модулите – разпределението на функционалността по модули, обекти, процедури, функции и връзки между тях;
 - Диаграма на процесите – процеси, нишки, синхронизация между тях, потоци от данни и събития, които ги свързват;
 - Схема на внедряването – процесори, устройства, сървъри, датчици, комуникационни устройства, разположението и връзките на софтуерните елементи с тях;

Стъпка #3 (прод.)

- Архитектурни похвати, схеми и тактики, както и качествените характеристики, които те преследват (3-9 слайда):
 - COTS (component-off-the-shelf) и тяхната интеграция (0-2 слайда);
 - Не повече от 3 от най-важните сценарии за употреба, вкл. ресурсите (1-3 слайда);
 - Не повече от 3 от най-важните сценарии за промяна, вкл. въздействието (1-3 слайда);
 - Архитектурни рискове относно изпълнението на изискванията (1-3 слайда);
 - Кратък терминологичен речник (0-1 слайда);

Стъпка #4

- Стъпка #4 – Идентифициране на архитектурния подход: Архитектът изброява използваните архитектурни стилове и схеми и обосновава тяхната употреба

Стъпка #5

- Създаване на **дърво на качествените атрибути**. Коренът на дървото е описание на качествените характеристики на системата – т.нар. Utility (практичност).
- Практичността се разделя на няколко основни качествени характеристики, като конкретните наименования не са толкова важни (напр. “бързодействие”);
- Всяка от качествените характеристики се разделя на конкретни изисквания (напр. “системата трябва да поддържа поне 20 fps”);
- Сценариите за качество са листата на дърветата. Сценариите тук са опростени – имат само по 3 елемента (стимул – събитие, който го генерира и какво бива стимулирано; състояние (какво се прави в момента); и резултат (измерваем, какво се случва в резултат на стимула)
- Обикновено дървото съдържа няколко десетки сценария, които трябва да се приоритизират, както по важност, така и по трудност

Стъпка #6

- Стъпка #6 – Анализ на архитектурния подход. Сценариите с най-висок приоритет от предишната стъпка се анализират един по един. Архитектът обяснява как архитектурата поддържа сценария. Екипът документира рискове, не-рискове, чувствителни сценарии и компромиси.
- Зад всичко това стои дискусия с архитекта, който трябва да защити позицията си. В следствие на дискусията анализът може да се задълбочи, като целта е екипа да бъде убеден в извлечените относно сценария детайли. А може и да не се задълбочи, ако архитектът все още няма виждане по някои от въпросите.
- На края на стъпка 6 екипът по оценка би следвало да има информация относно:
 - Най-важните аспекти на цялостната архитектура;
 - Обосновката на основните архитектурни решения;
 - Списък с рисковете, не-рисковете и компромисите;

Пауза

- Фаза 1 приключва със стъпка #6
- Следва пауза от 2-3 седмици
- През това време екипът по оценка обобщава наученото, неформално комуникирайки с архитекта (примерно по телефон, e-mail и т.н.), анализират се допълнителни второстепенни сценарии, изчистват се детайли, въпроси и т.н.
- Ръководителя на екипа по анализ запознава останалите заинтересовани лица със резултатите до тук и заедно с тях започва Фаза 2

Стъпка #7

- Стъпка #7 – брейнсторм и приоритизиране на сценариите. На тази стъпка останалите заинтересовани дискутират кои според тях са най-важните сценарии, като най-добре това се прави на т.н. брейнсторм сесия – събират се всички вкупом и си обменят мнения. На всеки се дава право да гласува за 30% от всички сценарии и тези с най-много гласове печелят
- Ако избраните сценарии съвпадат с тези от т. 5, значи има добро разбирателство между архитекта и заинтересованите лица. Ако има големи разлики – това само по себе си е риск за проекта

Стъпка #8

- Стъпка #8 – повтаря се стъпка #6, но за сценариите от стъпка #7. Където е възможно се използва досега натрупаната информация.

Стъпка #9

- Стъпка #9 – прави се обобщение на всичко научено дотук:
 - Документираните архитектурни подходи;
 - Сценариите и техния приоритет;
 - Дървото със сценариите от т. 5;
 - Рисковете и не-рисковете;
 - Чувствителните точки и компромисите;
- Рисковете се обобщават в теми
- За всяка тема екипът идентифицира бизнес целите, които биха могли да бъдат засегнати
- Така на пръв поглед чисто технически второстепенен проблем може да се окаже критичен за постигането на бизнес целите и съответния мениджър да бъде известен за това

АТАМ - обобщение

- Какво не е АТАМ:
 - НЕ Е оценка на изискванията – от него не става ясно дали ВСИЧКИ изисквания ЩЕ бъдат покрити. Това което става ясно е дали настоящата архитектура ще поддържа най-важните от тях
 - НЕ Е оценка на кода – обикновено се случва в ранните стадии на жизнения цикъл, преди още да се говори за код
 - НЕ включва същинско тестване на системата – по същите причини
 - НЕ Е прецизен инструмент – дава възможност за идентифициране на рискове, като се анализират компромисните и чувствителните точки. Всичко това зависи от знанията и уменията на оценителите, както и от представянето на самия архитект
 - НЕ дава стойността (\$) на рисковете. Това е предмет на друг вид анализ – СВАН (Cost Benefit Analysis Method)

АНАЛИЗ НА СОФТУЕРНИ АРХИТЕКТУРИ

Част 2 – СВАН

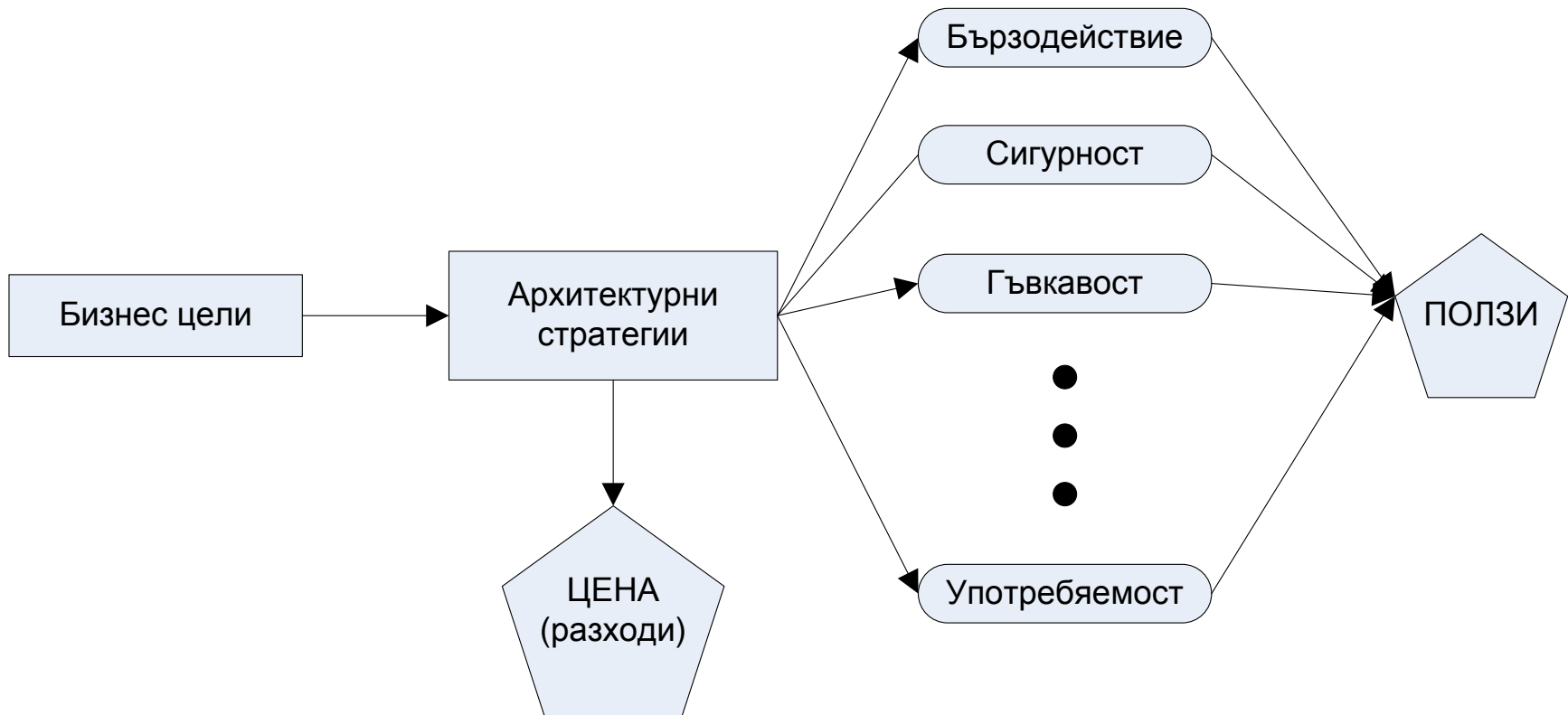
Мястото на СВAM

- В АТАМ става въпрос за анализ на компромиси, но всъщност най-важните компромиси се правят по икономически съображения
- АТАМ не отговаря на въпроса как дадената организация може да увеличи печалбите и да намали риска от даден софтуерен проект
- В миналото се е обръщало повече внимание на разходите (при това на преките такива), докато напоследък има тенденция да се отделя голямо внимание на ползите от дадена система
- Имайки предвид, че ресурсите необходими за построяването на СА са изчерпаеми, добре би било да има рационален процес, съгласно който да изберем един или друг архитектурен подход. Всички решения са свързани с определени разходи, допринасят с нещо (т.е. има полза от тях) и съдържат определен риск. Нужен е икономически модел, който да взема предвид тези фактори

Мястото на СВАМ

- На практика, СВАМ се базира на АТАМ и дава оценка на технико-икономическите аспекти на архитектурните решения
- Целта на архитекта е да увеличи разликата между ползите, извлечени от софтуерната система и разходите направени за производството ѝ
- СВАМ започва там, където АТАМ свършва и всъщност разчита на резултите от него

Мястото на СВAM



Мястото на СВAM

- Всяка софтуерна архитектура както техническа така и икономическа страна
- Икономическата страна на архитектурата е свързана както с цената на системата, така и с икономическия ефект от различните качествени произведени в следствие на различните качествени характеристики
- Връзките между бизнес цели, архитектурни решения и реализирани качествени характеристики се дава от ATAM
- СВAM се базира на разкритията относно тези връзки за да изгради представа за стойностите и ползите на всяко от решенията и за тяхната разлика – ROI (Return of Investment)

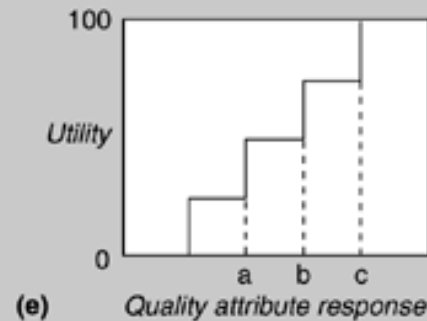
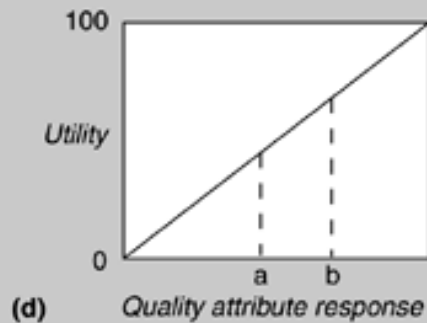
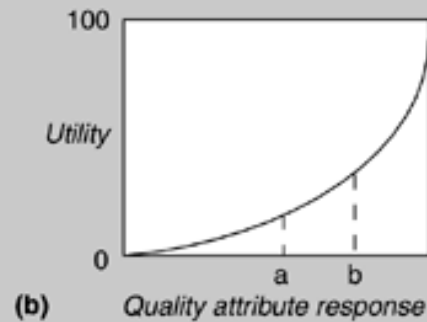
Основни идеи

- Методът се базира на оценка на полезността на взетите решения и оценка на тяхната цена
- За целта за всяко от качествата се разглеждат множество сценарии, като на всеки възможен техен резултат се присъжда степен на полезност (от страна на заинтересованите лица)
- Разглеждат се възможните архитектурни стратегии, които водят до различни вариации относно резултатите от сценариите
- Всяка стратегия си има цена и влияе върху едно или няколко качества, променяйки резултата от изпълненията им
- Отчетената полезност на всеки един от резултатите се сумира и така се формира окончателната ROI за съответния вариант на архитектурна стратегия

Вариации на сценариите

- В СВAM (подобно на ATAM), качествените характеристики на системата се описват чрез сценарии (с 3 елемента – стимул, обкръжение, резултат);
- За разлика от ATAM тук се разглеждат няколко сценария за една и съща характеристика, като на всеки резултат се присвоява степен на полезност

Криви на полезността (utility)



Криви на полезността

- Извличането на точната форма на кривата е дълъг и продължителен процес, но пък и това не е толкова необходимо
- Практиката показва, че 4 точки за сценарии са горе-долу достатъчни:
 - Най-добрия случай – (best-case scenario) – резултат, чието повишаване не води до повишаване на полезността. Полезност = 100
 - Най-лошия случай – (worst-case scenario) – разрешения санитарен минимум за системата – всеки по-лош резултат не е допустим за клиента. Полезност = 0
 - Настоящ резултат – сегашното състояние на нещата
 - Желан резултат – процент (може и 100%) от най-добрия
 - Може да има и 5^{та}, специфична за системата точка

Приоритизиране на сценариите за качество

- Различните сценарии имат различна значимост за различните заинтересовани лица
- За постигане на правилни резултати е редно да се постигне консенсус относно относителната тежест на сценариите
- Това става на две стъпки:
 - Заинтересованите гласуват за подредбата на сценариите по важност (базирани на очаквания резултат);
 - На така подредените сценарии се дава тежест, най-важния според общото мнение е оценен с 1, а останалите – с усреднена дробна стойност

Архитектурни стратегии

- Целта на архитекта е да се определи архитектурните стратегии, които да доведат нивото на резултата за всяка от качествените характеристики от “текущото” на “желаното”
- Част от СВAM се занимава и с това. За всяка стратегия можем да определим:
 - Най-вероятен резултат
 - Страничните ефекти на архитектурната стратегия върху другите качества
 - Оценка на стойността на съответната стратегия

Определяне на ползата и ROI на дадена стратегия

$$B_i = \sum_j (b_{ij} \cdot W_j)$$

$$R_i = \frac{B_i}{C_i}$$

b_{ij} е ползата, присвоена на стратегията i поради ефекта y върху сценарий j , като b_{ij} е се дава от съответната крива и е разликата между полезността на очакваното и настоящето ниво

W_j е тежестта, присъдена на сценария j

ROI се оценява като отношението на ползата към цената, като цената се определя по подходящ за целта начин. Стратегиите се подреждат по получените за ROI стойности

Формални стъпки на СВAM

(9 на брой)

- **Стъпка #1** – Сортиране на сценариите – събират се всички сценарии от ATAM, дава се възможност на заинтересованите лица да допринесат с още сценарии
- Всички събрани сценарии се приоритизират в зависимост от това как помагат за изпълняване на поставените бизнес цели
- За целите на по-нататъшния анализ се може да се изберат част от най-високо оценените сценарии

Стъпки на СВAM

- **Стъпка #2** – Рафиниране на сценариите – фокусира се вниманието върху резултата от сценариите, определят се най-добрия, най-лошия, текущия и желания резултати

Стъпки на СВAM

- **Стъпка #3** – Приоритизиране на сценариите – на всеки заинтересован се дават 100 гласа, които той да разпредели между сценариите, като се базира на желания резултат
- Резултатите се обобщават и се избират част от най-високо оценените от всички сценарии
- На най-високо оцененият сценарии се дава тежест 1, другите се оценяват от заинтересованите с дробен коефициент

Стъпки на СВAM

- **Стъпка #4** – Присвояване на полезност – на всеки от сценариите, останали след стъпка 3 се присвоява полезност за всяко от четирите нива на резултата

Стъпки на СВAM

- **Стъпка #5** – Разработка на архитектурните стратегии
– за всеки от сценариите се изработва архитектурна стратегия и се преценява очаквания резултат върху всеки от засегнатите сценарии

Стъпки на СВAM

- **Стъпка #6** – Оценка на полезността на очаквания от стратегиите резултат – за всеки засегнат сценарии чрез използване на интерполация се оценява полезността на очаквания резултат

Стъпки на СВAM

- **Стъпка #7** – Оценка на общата полезност на стратегията – извършва се сумиране на полезността за всеки засегнат от стратегията сценарии

Стъпки на СВAM

- **Стъпка #8** – Избор на стратегиите на база ROI – определя се цената на стратегията, съответно ROI. Стратегиите се подреждат съгласно коефициента на възвръщаемост и в зависимост от цената и др. ограничения (напр. във времето)
- Избират се онези най-полезни стратегии, които се побират в бюджетните и времевите ограничения

Стъпки на СВAM

- **Стъпка #9** – Потвърждаване на резултатите – по интуиция. За всяка от избраните стратегии се проверява дали съвпада с бизнес целите. Ако не съвпада – най-вероятно има нещо пропуснато по време на анализа
- Това става изцяло на база предишен опит на анализатора

ECS проектът на NASA

- Earth Observing System (EOS) е система от спътници, които доставят на НАСА информация за околната среда
- EOS Data Information System (EOSDIS) Core System (ECS) събира микро-данните от различните сателити, обработва ги и ги разпространява в добре познат формат на заинтересовани организации
- Целта е да има единен начин за съхранение (следователно и обработка) на данните, както и публично известен механизъм за предоставяне на нови полета и формати
- Събират се няколко стотин GB всеки ден
- С тях се извършват около 250 стандартни сметки, което води до създаването на няколко хиляди GB информация в 8 центъра за данни на територията на САЩ
- Системата има важни изисквания за бързодействие и надеждност. Имайки предвид и очакваното дълго време за експлоатация, гъвкавостта също се явява важно изискване

ECS проектът на NASA

- Ръководителят на проекта има много предложения за подобряване на системата (от предишен ATAM) и съответни бюджетни ограничения
- Задачата е да се изберат тези 10-20% от предложените архитектурни стратегии, които ще имат най-голям ефект върху системата и които ще се поберат в годишния бюджет
- В примера е даден анализа само на една от под-системите – DWAG (Data Access Working Group)

DWAG Стъпка #1 – Сортиране на сценариите

1	Да се намалят грешките при дистрибуция, които водят до увисване на процесите и ръчна намеса
2	Да се намалят грешките при дистрибуция, водещи до загуба на заявки
3	Да се намали броя на поръчките, които се провалят по време на заявяването
4	Да се намали броя на поръчките, които при провал водят до увисване на процеси и ръчна намеса
5	Да се намали броя на поръчките, които завършват с изгубване на поръчката
6	Няма добър метод за проследяване на пропадналите/отменените ECSGuest поръчки без ръчна намеса (напр. excel таблици)
7	Потребителите се нуждаят от повече информация за причината за грешка
8	Броят и размерът на поръчките се ограничава изкуствено поради софтуерни ограничения
9	Малките поръчки водят до голям брой известия до потребителите
10	Системата трябва да обработва поръчки от 50GB за ден, а от 1TB – за седмица

DWAG Стъпка #2 – Рафиниране на сценариите

Сценарий	Резултат			
	Най-лош	Текущ	Желан	Най-добър
1	10% увиснали	5%	1%	0%
2	>5% изгубени	<1%	0%	0%
3	10% провалени	5%	1%	0%
4	10% увиснали	5%	1%	0%
5	10%	<1%	0%	0%
6	изгубени 50% искат F1	25%	0%	0%
7	10% получават	50%	100%	100%
8	50%	30%	0%	0%
9	огранич. 1 за гранула	1	1/100	1/1000
10	< 50% вярно	60%	80%	>90%

DWAG Стъпка #3 – Приоритизиране на сценариите

Сценарий (Гласове)	Резултат			
	Най-лош	Текущ	Желан	Най-добър
1 (10)	10%	5%	1%	0%
2 (15)	увиснали > 5%	< 1%	0%	0%
3 (15)	изгубени 10%	5%	1%	0%
4 (10)	провалени 10%	5%	1%	0%
5 (15)	увиснали 10%	< 1%	0%	0%
6 (10)	изгубени 50% искат F1	25%	0%	0%
7 (5)	10%	50%	100%	100%
8 (5)	получават 50% огранич.	30%	0%	0%
9 (10)	1 за гранула	1	1/100	1/1000
10 (5)	< 50% вярно	60%	80%	> 90%

DWAG Стъпка #4 – Присвояване на ПОЛЕЗНОСТ

Сценарий	Гласове	Полезност/резултат			
		Най-лош	Текущ	Желан	Най-добър
1	10	10	80	95	100
2	15	0	70	100	100
3	15	25	70	100	100
4	10	10	80	95	100
5	15	0	70	100	100
6	10	0	80	100	100
7	5	10	70	100	100
8	5	0	20	100	100
9	10	50	50	80	90
10	5	0	70	90	100

DWAG Стъпка #5 – Разработка на стратегии

Номер на стратегия	Описание на стратегията	Повлияни сценарии	Текущ резултат	Очакван резултат
S1	Своевременно записване на поръчките	3 5 6	5% пропаднали <1% изгубени	2% 0% 0%
S2	Разбиване на поръчки	8	25% искат F1 ограничени	15% ограничени
S3	Обединяване на поръчки	9 10	1 60% вярно	1/100 55%
S4	Частично изпълняване на поръчките	4	5% увисват	2%
S5	Ръчна промяна в поръчките	1	5% увисват	2%

DWAG Стъпка #5 – Разработка на стратегии (прод.)

Номер на стратегията	Описание на стратегията	Повлияни сценарии	Текущ резултат	Очакван резултат
S6	Повторна обработка на поръчка	1	5% увисват	2%
S7	Ръчно приключване на поръчка	1	5% увисват	3%
S8	Операторски контрол върху процеса на уведомление	6 7	25% искат F1 50% получават	20% 90%
S9	Статус на всеки ред от	6 7	25% искат F1 50%	10% 95%
S10	поръчката достъп до досието на	7	50% получават получават	60%

клиента

DWAG Стъпка #6 – Оценка на полезността на стратегиите

Стратегия	Повлияни сценарии	Текуща полезност	Очаквана полезност
S1	3	70	90
	5	70	100
	6	80	100
S2	8	20	60
S3	9	50	80
	10	70	65
S4	4	80	90
S5	1	80	92
S6	4	80	85
S7	1	80	87
S8	6	80	85
	7	70	90
S9	6	80	90
	7	70	95
S10	7	70	75

DWAG Стъпка #7 – Оценка на общата полезност

Стратегия	Сценарии	Тежест	Необработена полезност	Умножена Полезност	Крайна полезност
S1	3	15	20	300	950
	5	15	30	450	
	6	10	20	200	
S2	8	5	40	200	200
S3	9	10	30	300	275
	10	5	-5	-25	
S4	4	10	10	100	100
S5	1	10	12	120	120
S6	4	10	5	50	50
S7	1	10	7	70	70
S8	6	10	5	50	150
	7	5	20	100	
S9	6	10	10	100	225
	7	5	25	125	
S10	7	5	5	25	25

DWAG Стъпка #8 – Избор на стратегиите на база ROI

Стратегия	Стойност	Полезност	ROI	Класиране
S1	1200	950	0.79	1
S2	400	200	0.5	3
S3	400	275	0.69	2
S4	200	100	0.5	3
S5	400	120	0.3	7
S6	200	50	0.25	8
S7	200	70	0.35	6
S8	300	150	0.5	3
S9	1000	225	0.22	10
S10	100	25	0.25	8

Обобщение - СВAM

- Методът е итеративен и често се базира на неформална комуникация с заинтересованите лица, които на базата на консенсус вземат решения за полезността и цената на дадените стратегии
- Естествено, това отнема много време и е необходимо гъвкавото му приложение в зависимост от обстановката
- Въпреки това, при големи проекти икономическият ефект от приложението на структуриран метод за анализ е винаги положителен в сравнение с ad hoc анализа и вземането на решения по метода “гледане в точка”